

Apresentação e Plano de Testes

Motivação para implementação projeto

As cadernetas de vacinas em papel deterioram mais rapidamente, passíveis de perda, além da facilidade destrutiva total ou parcialmente se eventualmente molhadas, ou a própria erosão do tempo. Além disso, os primeiros anos de vida com um bebê e seu crescimento, são intensos com nova rotina na vida da família e não é incomum passar do prazo das vacinas necessárias, e passado o prazo podem ficar sem a imunização tão importante para o desenvolvimento da criança, reduzindo acometimento por doenças potencialmente fatais como a meningite, sarampo, poliovírus selvagem, etc.

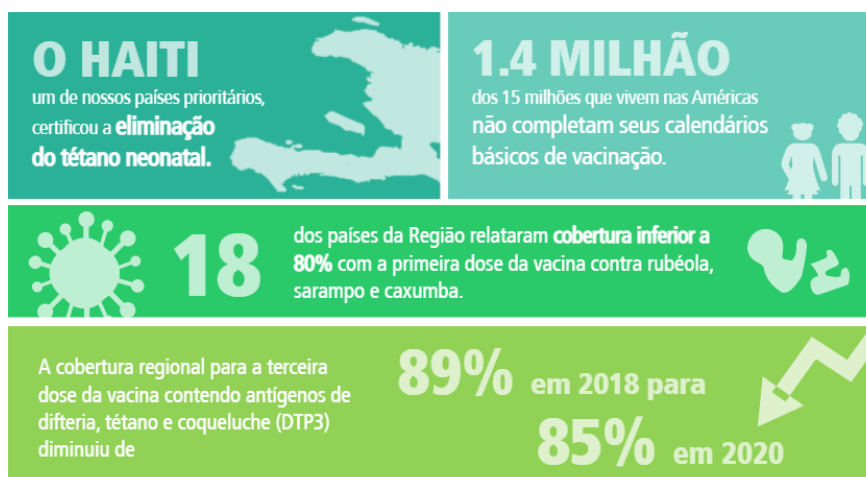
Justificativa para o projeto

O projeto de caderneta de vacina digital foi desenvolvido pensando na proteção à saúde e bem-estar das crianças de 0 a 10 anos de idade, tendo como objetivo principal o apoio aos pais ou responsáveis legais desse público, e traz facilidades às famílias com a possibilidade de concentrar todas as informações referentes aos imunizantes (vacinas) adotados pelo Ministério da Saúde do Brasil, em um só local, recebem alertas próximas as datas e campanhas, acompanhando as tendências tecnológicas, de forma segura, privada e prática, além de contribuir com desenvolvimento ecológico e sustentável, com a redução de papéis no planeta (segundo estudos da WWF, a produção de uma tonelada de papel consome de 50 a 60 eucaliptos, 100 mil litros de água e 5 mil KW/h de energia).

“Muitos pais nunca ouviram falar de pólio, rubéola e difteria. Por essa razão, não levam os filhos para se proteger”, nota o pediatra Juarez Cunha, presidente da Sociedade Brasileira de Imunizações. Além disso, nosso calendário é bem completo — ao todo, são 19 imunizantes, que previnem 28 doenças — que, só no primeiro ano de vida da criança, os pais são obrigados a ir ao posto nove vezes. “Alguns se esquecem ou deixam para depois”, relata Cunha.

<https://saude.abril.com.br/medicina/por-que-as-pessoas-estao-tomando-menos-vacina/>

Vacina no mundo.



Definição de Testes

Os testes de Software almejam validar a eficácia, resiliência e usabilidade da plataforma antes do seu lançamento oficial. Assim certificarmos-nos que o software atende aos requisitos funcionais e de desempenho idealizados que motivaram seu desenvolvimento. Com os dados obtidos identificar e corrigir quaisquer ‘bugs’ ou problemas no sistema. Verificar a usabilidade e a eficiência do software no gerenciamento de chamados e incidentes.

Abordagem

Nosso projeto utiliza a linguagem Gherkin, assim como as ferramentas Cucumber para automação, que segue a abordagem de Desenvolvimento Orientado a Comportamento (BDD - Behavior Driven Development) e JUnit para efetuar anotações dos testes e gerar as métricas necessárias.

Objetivo da Avaliação e Estimulo dos Testes

O objetivo dos testes nesta iteração é garantir a entrega do Software Carteira de Vacinas com alta consistência dos dados apresentados ao usuário, proporcionando usabilidade fluida com integridade de dados, assim como conectividade, e interação com o Sistema Único de Saúde (SUS) do Brasil. Nosso foco ímpar em assegurar a esse requisitos junto a segurança do sistema, proporciona experiência de usuário e contribui para o sucesso do projeto.

Nosso estímulo para os testes é o impulso pela busca incessante de qualidade do nosso software. É a convicção de que testar minuciosamente cada aspecto não apenas garante funcionalidade, mas também fortalece a confiança operacional do produto. Com as devidas verificações, temos a oportunidade de aprimorar, mitigar riscos e superar algumas expectativas.

Motivadores dos Testes

Riscos de Qualidade:

Identificar e mitigar riscos que possam comprometer a qualidade do software, dessa forma podemos garantir que o produto final atenda aos padrões esperados de desempenho e confiabilidade esperados.

Riscos Técnicos:

Identificar os desafios técnicos que podem surgir durante a implementação do software, assegurando que a solução seja robusta e escalável.

Riscos do Projeto:

Nessa etapa podemos avaliar e minimizar riscos associados à entrega do projeto, garantindo o cumprimento dos prazos e que o software atenda às expectativas levantadas pelos stakeholders.

Casos de Uso:

Validar e certificar que as funcionalidades do software em relação aos casos de uso identificados, garantam que todas as interações previstas estejam implementadas corretamente com excelência.

Requisitos Funcionais:

Verificar se o software atende a todos os requisitos funcionais especificados, garantindo que as principais funcionalidades estejam implementadas de acordo com as expectativas.

Requisitos Não Funcionais:

Certificar que os aspectos não funcionais, como desempenho, segurança e usabilidade, para garantam que o software funcione corretamente, e também atenda aos requisitos de desempenho e experiência do usuário.

Experiência do Usuário:

Garantir que a experiência do usuário seja positiva, quanto a usabilidade, a eficiência na execução de tarefas e a navegabilidade da interface.

Casos de Teste:

- Fazer o levantamento preciso quando ao funcionamento do Login do software está funcionando corretamente, cadastrando novos usuários corretamente, e se os mesmos seguem os padrões, pré determinados no projeto;
- Verificar se o retorno das informações disponibilizadas cadastradas, retorna seguindo os padrões SUS;
- Desenvolver casos de teste que cubram todos os requisitos funcionais do software:
- Relatório de vacinas, Datas de aplicação, Locais de aplicação da vacina;
- Permitir que o usuário cadastre as aplicações realizadas pelo mesmo;
- Permitir que o Software, gere alertas das próximas vacinas a serem aplicadas;
- Permitir que o usuário localize locais de vacinação;
- Permitir que o usuários, busque vacinas obrigatórias para onde o mesmo queira viajar;

Testes de Usabilidade

Verificar a usabilidade do software, verificando a facilidade de uso e navegabilidade da interface do usuário, e se todos os comandos respondem corretamente.

Teste de Conexão com Banco de Dados

Testa se a conexão com o banco de dados foi bem estabelecida corretamente;

Gherkin:

- Funcionalidade: conferir conexão com DB;

- Primeiro Cenário:

- verificar conexão com banco com erro;

- Se dado o usuário o software chamou DbConnection;

- Quando chamar a função verificarStatus;

- Então recebe 'false' para conexão;

- Segundo Cenário:

- verificar conexão com banco com sucesso;

- Se dado o usuário chamou DbConnection;

- Quando chamar a função verificarStatus

- Então recebe 'true' para conexão;